Wykład 4:
Protokoły TCP/UDP i usługi sieciowe

Adres aplikacji: numer portu

- Protokoły w. łącza danych (np. Ethernet) oraz w. sieciowej (IP) pozwalają tylko na zaadresowanie komputera (interfejsu sieciowego), a komunikacja zachodzi pomiędzy procesami (aplikacjami)
- Do adresowania wybranej aplikacji stosuje się numer portu – adres warstwy transportowej (TCP)
- Numer portu jest dwubajtowy (0-65535)
- Według specyfikacji URL numer portu dodajemy po nazwie hosta i dwukropku
- Pominięcie numeru portu oznacza, że zostanie użyty numer portu domyślny dla danej usługi
- Łącze wymaga numerów portów po obu stronach

Dobrze znane porty

- Najpopularniejsze usługi mają przydzielone specjalne porty, na których standardowo nasłuchują ich procesy serwerów
- Organizacja IANA utrzymuje listę portów przydzielonych standardowym usługom:
  `http://www.iana.org/assignments/port-numbers`
- Lista jest lokalnie zapisana w `/etc/services`
- Porty 1-1023 są w Linuxie zarezerwowane i mogą być przydzielone tylko przez użytkownika `root`
- Porty protokółów TCP i UDP są niezależne
- Nawiązując połączenie musimy również podać numer portu na swojej maszynie – zwykle jest to „wysoki” numer portu

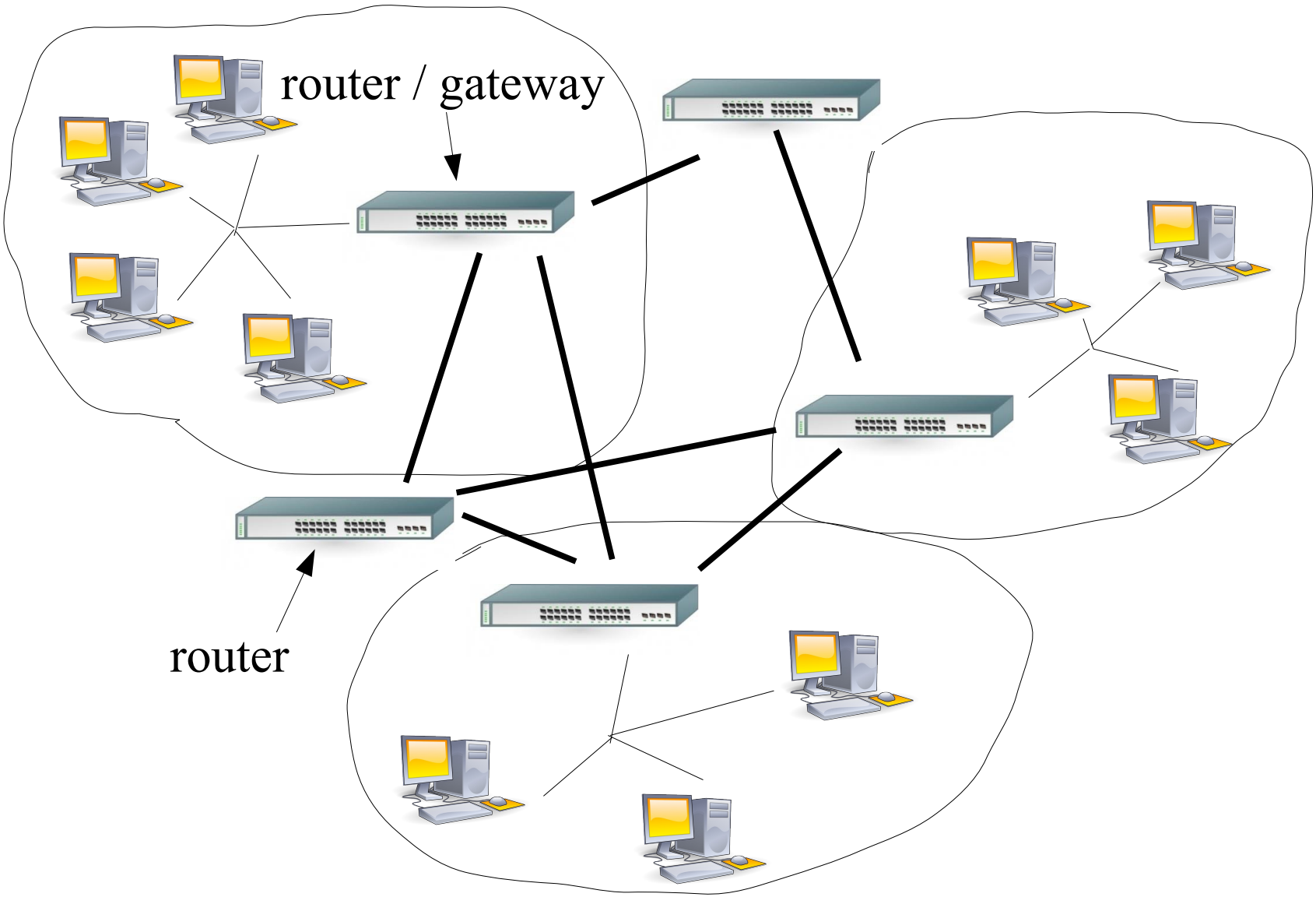
Usługi w. transportowej a w. sieci

- Usługi w. transportowej są ograniczane możliwościami w. sieci (przepustowość, opóźnienie), ale mogą (nie muszą) oferować jakościowo nową funkcjonalność
  - Niezawodność połączenia (izoluje proces komunikacji od zawodności w. sieci)
  - Oferowanie połączenia logicznego (mimo iż komunikacja w w. sieci jest pakietowa)
  - Oferowanie bezpieczeństwa komunikacji, np. poprzez szyfrowanie danych, mimo iż pakiety podróżują przez nieznane systemy
  - Kontrola przeciążenia łącza
  - Multipleksowanie/demultipleksowanie

Protokoły TCP i UDP

- Protokół UDP (User Datagram Protocol):
  - Usługa: bezpołączeniowe dostarczenie datagramów
  - Nie kontroluje poprawności i prędkości przesyłu
  - Minimalizuje przesył dodatkowych danych
- Protokół TCP (Transfer Control Protocol)
  - Usługa: połączeniowe, niezawodne przesyłanie danych (strumień)
  - Odpowiada za segmentowanie i powtórne scalanie danych, kontroluje prędkość przesyłu
  - Wykorzystuje potwierdzenie z retransmisją (wysyła dane aż do otrzymania potwierdzenia)

Połączenie logiczne



- W. transportowa - logiczne połączenie aplikacji
- Działa na systemach końcowych, nie routerach!

Architektura połączenia

- Połączenie pomiędzy aplikacjami odbywa się zwykle w architekturze klient-serwer (TCP/UDP)
  - Jedna aplikacja oczekuje na połączenie (serwer), druga je inicjuje (klient)
  - Zakłada się, że transfer danych będzie w większości przebiegał od serwera do klienta, klienci nie komunikują się ze sobą
- Możliwa jest inna architektura: symetryczne połączenie pomiędzy równorzędnymi hostami (często UDP ale możliwe też TCP)
  - Wykorzystanie: sieci P2P, telefonia internetowa
  - Połączenie takie można traktować jako złożenie dwóch połączeń klient-serwer

TCP czy UDP? Jak wybrać?

- Protokoły TCP i UDP uzupełniają się, wybieramy jeden z nich w zależności od potrzeb
- TCP
  - Gdy zależy nam na pewności połączenia i nie tolerujemy utraty danych
  - Cena: mniejsza wydajność wykorzystania łącza
  - Połączenie jest automatycznie ograniczane gdy rośnie zajętość łącza
  - Odpowiednie dla: poczta, zdalne logowanie, WWW, transfer plików
- UDP
  - Gdy zależy nam na maksymalnej wydajności i tolerujemy utratę danych (albo sami implementujemy niezawodność w warstwie wyższej!)
  - Cena: bardziej skomplikowana implementacja
  - Nie podlegamy automatycznemu ograniczaniu prędkości (dobrze dla nas, źle dla Internetu ...)
  - Odpowiednie dla: strumieniowe multimedia, telefonia VOIP, routing, DNS, sieci P2P

Polecenie `netstat`

- Informuje o stanie podsystemu sieciowego
- Z opcją `-nap` wypisuje wszystkie wykorzystywane porty, wraz z podaniem ich stanu, z opcją `--ip` wypisuje tylko porty TCP/UDP
- Umożliwia wypisanie aktualnej tablicy routingu (opcja `-r`)
- Pokazuje statystyki wykorzystanie interfejsów sieciowych (opcja `-i`)
- Oraz protokołów (opcja `-se`)
- Można go użyć w trybie ciągłym, w którym na bieżąco wyświetlane są żądane dane

Firewall i gateway

- Firewall to urządzenie działające w w. transportowej, kolejne w hierarchii urządzeń sieciowych
  - Musi rozpoznawać nie tylko numery IP, ale także numery portów TCP/UDP
  - Może sterować ruchem (przepuszczać/blokować lub sterować prędkością) na podstawie numerów portów (w domyśle – usług)
  - Realizacja: filtrowanie pakietów
- Gateway to urządzenie działające w w. aplikacji
  - Do przepuszczenia połączenia wymagana jest autentykacja na urządzeniu
  - Zwykle granulacja jest na poziomie usług (kolejna usługa wymaga nowego logowania)

Serwer proxy

- Serwer proxy to kolejny sposób dostępu do Internetu z jednoczesną izolacją sieci lokalnej
  - Dla każdej usługi na serwerze proxy uruchamiany jest oddzielny program, który przekazuje i jednocześnie buforuje komunikaty przekazywane pomiędzy sieciami
  - Może być tradycyjny (wymaga logowania) lub przezroczysty („transparent proxy”)
  - Najczęściej łączony z technologią NAT - adresy w sieci lokalnej są nieroutowalne
  - Pakiety nigdy nie przechodzą pomiędzy sieciami, są zawsze buforowane i ponownie wysyłane przez serwer proxy